

Индивидуальный предприниматель Селендеева О.Н.

УТВЕРЖДАЮ

Индивидуальный предприниматель

_____/Селендеева О.Н.//

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«РАЗРАБОТКА 2D И 3D ИГР НА PYTHON»**

Москва, 2023

Оглавление

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
1.1. Общая характеристика программы	3
1.2. Цели и задачи программы	4
7. Научить создать собственные 2D или 3D игры в среде Godot и публиковать их в интернет.	5
1.3. Планируемые результаты обучения	5
2. УЧЕБНЫЙ ПЛАН ПРОГРАММЫ	6
3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК	8
4. РАБОЧАЯ ПРОГРАММА	1
5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ	7
5.1. Контроль знаний, умений и навыков	7
5.2. Критерии оценивания освоения программы при проведении различных форм контроля:	7
6. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ (УЧЕБНИКИ, РАЗДАТОЧНЫЕ МАТЕРИАЛЫ, МЕТОДИЧЕСКИЕ УКАЗАНИЯ, ПЛАКАТЫ, СЛАЙДЫ, ИНТЕРНЕТ-РЕСУРСЫ)	9
6.1. Информационные и учебно-методические условия реализации программы	9
6.2. Рекомендованная литература, рекомендованная для освоения программы:	9
6.3. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", рекомендованных для освоения программы	10
7. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ	11
7.1. Материально-техническая и ресурсная база	11
7.2. Кадровое обеспечение программы	11

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1. Общая характеристика программы

Данный документ описывает комплекс основных характеристик образования (объем, содержание, планируемые результаты) и организационно-педагогических условий, который представлен в виде учебного плана, календарного учебного графика, рабочих программ учебных предметов, модулей, иных компонентов, а также оценочных и методических материалов общеобразовательной общеразвивающей программы "Разработка 2D и 3D игр на Python".

В ходе дополнительной общеобразовательной общеразвивающей программы "Разработка 2D и 3D игр на Python" обучающиеся научатся самостоятельно ориентироваться в среде разработки Godot, овладеют навыками создания сюжета, дизайна и механик компьютерных игр, научатся создавать персонажей, использовать подходы, принципы и технологии создания современных двумерных и трёхмерных игр с использованием языка программирования Python.

Выдача обучающимся документов о дополнительном образовании (сертификат установленного образца) осуществляется при условии успешного прохождения итоговой аттестации.

Программа разработана на основе следующих **нормативных документов**:

- Федеральный закон от 29 декабря 2012 г. №273-ФЗ «Об образовании в Российской Федерации»;
- Приказ Министерства просвещения РФ от 9 ноября 2018 г. N 196 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам"
- Постановление Главного государственного санитарного врача РФ от 28.09.2020 № 28 «Об утверждении санитарных правил СП 2.4.3648-20 "Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи" (вместе с "СП 2.4.3648-20. Санитарные правила...")» (Зарегистрировано в Минюсте России 18.12.2020 № 61573).

Направленность (профиль) программы: техническая.

Актуальность программы, соответствие государственному, социальному заказу/запросам.

В современном мире умение программировать является ценным и востребованным навыком, который все больше и больше ценится на рынке. В результате прохождения обучения слушатель сформирует базу знаний в области программирования в среде разработки Godot, активизирует навыки использования полученных знаний и умений в практической деятельности, разовьет логическое мышление, проявит интерес к программированию, раскроет свои способности в сфере разработки игр. Обучающимся будут созданы оптимальные условия для всестороннего удовлетворения потребностей и развития их индивидуальных склонностей и способностей, появится мотивация личности к познанию и техническому творчеству. Полученные на данной программе навыки помогут сделать первые шаги в таких востребованных профессиях, как программист, геймдизайнер, разработчик игр.

Отличительные особенности программы: по окончании обучения учащиеся с помощью полученных на курсе знаний и навыков создадут несколько собственных 2D и 3D-игр, создадут портфолио из них. На занятиях ребенок погружается в мир IT, осваивает азы программирования через игру. Обучение проходит в среде разработки Godot, с использованием языка программирования Python, и начинается с самых базовых понятий, поэтому курс подходит для ребят, не имеющих опыта в программировании.

Срок обучения: программа реализуется в объеме 84 академических часов, 32 недель (8 месяцев).

Режим занятий: 2-4 академических часа в неделю

Продолжительность академического часа – 45 минут.

Занятия начинаются не ранее 9.00 часов утра и заканчиваются не позднее 20.00 часов. Для обучающихся в возрасте 16-18 лет допускается окончание занятий в 21.00 часов.

Продолжительность занятий в учебные дни - не более 3-х академических часов в день, в выходные и каникулярные дни - не более 4 академических часов в день. После 30-45 минут теоретических занятий организуется перерыв длительностью не менее 10 мин.

Адресат программы и примерный портрет слушателя курсов: программа разработана для учащихся от 12 до 16 лет, которых интересуют современные технологии создания компьютерных игр. К освоению дополнительной общеобразовательной программы – дополнительной общеразвивающей программы допускаются: лица без предъявления требований к уровню образования.

По завершении реализации программы, как правило, проводится анкетирование обучающихся с целью изучения мнения по вопросу эффективности и информативности проведенного обучения, уровню организации учебного процесса, удовлетворенности учебно-методическим материалом, работниками образовательной организации проводится анализ высказанных предложений и пожеланий.

1.2. Цели и задачи программы

Цель программы – знакомство с фундаментальными структурами языков программирования, получение визуального представления о программировании игр, получение навыков создания сюжета, дизайна и механик игры, формирование интереса к увлечению программированием и раскрытию своих способностей в сфере IT-технологий.

Задачи программы:

1. Сформировать системные знания в области компьютерных технологий и разработки игр и программ.
2. Предоставить специализированные практические навыки по основам программирования.
3. Сформировать навыки игрового программирования, применения подходов и технологий создания современных игр.
4. Научить создавать собственные игры в среде разработки Godot.
5. Дать представление о профессии разработчика игр.
6. Сформировать навыки работы над проектами, развивая творческий подход, а также навыки тайм-менеджмента, решения задач и принятия решений.

7. Научить создать собственные 2D или 3D игры в среде Godot и публиковать их в интернет.

1.3. Планируемые результаты обучения

По итогам освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы "Разработка 2D и 3D игр на Python" обучающиеся должны будут овладеть следующими знаниями, умениями и навыками:

Знать:

- основы работы в среде разработки Godot
- базовый синтаксис языка программирования Python
- способы работы с переменными и условиями,
- основные алгоритмы программирования,
- использование системы 2D и 3D координат в игровом программировании,
- понятие и основы работы с префабами,
- основы работы с пользовательским интерфейсом,
- принципы и технологию создания современных двумерных и трёхмерных игр.

Уметь:

- самостоятельно ориентироваться в среде разработки Godot,
- использовать в работе принципы написания кода и создания алгоритмов,
- применять творческий подход к реализации задач в игре,
- создавать игровых и неигровых персонажей,
- добавлять в игру модели объектов,
- работать с текстурами объектов,
- создавать активные компоненты игры,
- добавлять пользовательский интерфейс,
- настраивать анимацию персонажей,
- работать с различными материалами и источниками света,
- писать простые скрипты на языке Python и настраивать их взаимодействие,
- создавать свои игры, включая их подготовку, сборку и тестирование,
- работать на результат,
- создавать из созданных игр сайт-портфолио,
- создавать и презентовать свои проекты.

Владеть навыками в области:

- создания сюжета, дизайна и механик игры,
- реализации концепции уровня игры,
- автоматизации игровых процессов,
- синхронизации игры и звуков,
- разработки компьютерных игр,
- поиска ошибок и улучшения написанного кода,
- тайм-менеджмента, постановки и решения задач и принятия решений,
- презентации проектов и портфолио.

2. УЧЕБНЫЙ ПЛАН ПРОГРАММЫ

В процессе преподавания курса "Разработка 2D и 3D игр на Python" используются как классические методы обучения (лекции), так и различные виды практической работы обучающихся по заданию преподавателя, которые направлены на развитие навыков создания 2D и 3D игр, креативных качеств и на поощрение интеллектуальных инициатив учащихся.

Формы организации образовательного процесса (индивидуальные, групповые и т.д.) и другие виды занятий по программе определяются содержанием программы. Образовательная деятельность обучающихся предусматривает следующие виды учебных занятий и учебных работ: лекции, практические и семинарские занятия, круглые столы, мастер-классы, тренинги, семинары по обмену опытом, проведение открытых занятий, консультации и другие виды учебных занятий и методической работы, определенные учебным планом.

№	Наименование дисциплины/раздела/темы	Количество академических часов				Форма аттестации /контроля
		Всего	в т.ч. аудиторных		СРС	
			теория	практич. занятия		
1	Знакомство с Godot и основами интерфейса	2	1	1	0	Практическое задание
2	Создание 2D-игры Платформер	3	1	1	1	Практическое задание
3	Монеты и UI	4	1,5	1,5	1	Практическое задание
4	Враги и порталы	3	0,5	1	1,5	Практическое задание. Контрольные вопросы
5	Пули и жизни	2	0,5	0,5	1	Практическое задание
6	Зелья и powerups	2	1	1	0	Практическое задание
7	Предметы и инвентарь	2	1	1	0	Практическое задание
8	NPC и диалоги	3	1	1	1	Практическое задание. Контрольные вопросы
9	Знакомство с 3D	3	1	1	1	Практическое задание
10	Физика в Godot	2	0,5	0,5	1	Практическое задание
11	Terrain. Формирование ландшафта	2	0,5	0,5	1	Практическое задание
12	NPC и враги в 3D	4	1	1	2	Практическое задание. Контрольные вопросы
13	Создание оружия и пуль	2	0,5	0,5	1	Практическое задание
14	UI в 3D играх	2	1	1	0	Практическое задание

15	Добавление звукового оформления в игру	2	1	1	0	Практическое задание
16	Визуальное оформление и работа со светом	3	1	1	1	Практическое задание. Контрольные вопросы
17	Подготовка к созданию хоррора	3	1	1	1	Практическое задание
18	Создание хоррора	2	0,5	0,5	1	Практическое задание
19	Доработка хоррора	2	0,5	0,5	1	Практическое задание
20	Улучшение хоррора	4	1	1	2	Практическое задание. Контрольные вопросы
21	Подготовка к созданию игры Гонки	2	0,5	0,5	1	Практическое задание
22	Физика и UI в гонках	2	1	1	0	Практическое задание
23	Поведение соперников и финиш	2	1	1	0	Практическое задание
24	Улучшение игры гонки	4	1	1	2	Практическое задание. Контрольные вопросы
25	Возвращение в 2D и создание игры Roguelike RPG	3	1	1	1	Практическое задание
26	Знакомство с Tileset'ами	2	0,5	0,5	1	Практическое задание
27	Процедурная генерация	2	0,5	0,5	1	Практическое задание
28	Добавление сундуков и собираемых предметов	4	1	1	2	Практическое задание. Контрольные вопросы
29	Добавление врагов и системы жизней	3	1	1	1	Практическое задание
30	Новые локации	2	0,5	0,5	1	Практическое задание
31	Квесты	2	0,5	0,5	1	Практическое задание
32	Экспорт игры и выкладывание на хостинг. Подготовка к демонстрации. Итоговая аттестация-презентация	4	1	1	2	Практическое задание. Контрольные вопросы. Защита проекта
ИТОГО		84	26,5	27	30,5	

3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Календарный учебный график формируется при осуществлении обучения в течение всего календарного года. По мере набора групп слушателей по программе составляется календарный график, учитывающий объемы лекций, практики, самоподготовки, другие формы организации занятий.

Темы / недели	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	Итого часов
1. День первый Знакомство с Godot и основами интерфейса	2																																2
2. Создание 2D-игры Платформер		3																															3
3. Монеты и UI			4																														4
4. Враги и порталы				2,5																													2,5
Промежуточный контроль				0,5																													0,5
5. Пули и жизни					2																												2
6. Зелья и powerups						2																											2
7. Предметы и инвентарь							2																										2
8. NPC и диалоги								2,5																									2,5
Промежуточный контроль								0,5																									0,5
9. Знакомство с 3D									3																								3
10. Физика в Godot										2																							2
11. Terrain. Формирование ландшафта											2																						2
12. NPC и враги в 3D												3																					3
Промежуточный контроль												1																					1
13. Создание оружия и пуль													2																				2
14. UI в 3D играх														2																			2
15. Добавление звукового оформления в игру															2																		2
16. Визуальное оформление и работа со светом																2,5																	2,5
Промежуточный контроль																0,5																	0,5
17. Подготовка к созданию хоррора																	2																2
18. Создание хоррора																		2															2

4. РАБОЧАЯ ПРОГРАММА

1. День первый Знакомство с Godot и основами интерфейса

- изучение основных свойств и особенностей Godot
- изучение интерфейса Godot
- знакомство с системой Нод
- добавление картинок в проект
- создание объектов в Godot

Практическое задание: добавить новые объекты на уровне

2. Создание 2D-игры «Платформер»

- изучение основ программирования в Godot
- создание первых скриптов
- добавление ноды «CollisionShape»
- создание основы игры «Платформер»

Практическое задание: придумать и реализовать концепцию уровня

3. Монеты и UI

- знакомство с переменными и условиями
- создание скрипта монет
- изучение ноды Area2d и соответствующих событий
- знакомство с пользовательским интерфейсом и создание элементов Text
- изучение команд для взаимодействия с UI

Практическое задание: добавление нового предмета для сбора

4. Враги и порталы

- изучение работы с деревом и командой change_scene
- создание скриптов для движения и атаки врага
- знакомство с понятием префаб
- создание новых уровней

Практическое задание: создание объектов из префаба, написание скриптов движения камеры.

5. Пули и жизни

- углубленное изучение работы переменных
- добавление новых элементов в UI
- создание скриптов для вывода значений в UI
- знакомство с instance и созданием объектов через код

Практическое задание: создать предметы, увеличивающие количество жизней и пуль.

6. Зелья и powerups

- изучение работы взаимодействия скриптов между собой
- создание в игре зелий и powerup'ов
- знакомство с Timer'ом и командами ожидания

Практическое задание: добавить новый вид зелий в игру

7. Предметы и инвентарь

- знакомство с созданием массивов
- добавление в игру предметов
- знакомство с элементами grid, texture button
- изучение событий нажатия кнопок

Практическое задание: добавить новые предметы для сбора

8. NPC и диалоги

- установка расширений в godot
- знакомство со словарями
- создание неигровых персонажей
- создание внутриигрового магазина

Практическое задание: добавление новых диалогов и NPC в игру

9. Знакомство с 3D

- знакомство с управлением в 3D-пространстве
- создание примитивных объектов в godot
- скачивание и импорт моделей
- изучение координат в Godot
- создание материалов.

Практическое задание: добавить и импортировать в игру новые модели

10. Физика в Godot

- изучение нод StaticBody и KinematicBody
- создание скрипта движения персонажа
- создание и дизайн игры Паркур

Практическое задание: улучшить внешний вид игры, добавить новые игровые элементы

11. Terrain. Формирование ландшафта

- установка плагинов в Godot
- знакомство с возможностями Terrain Editor
- создание 3D-карты мира
- добавление персонажа

Практическое задание: сделать игровой мир более живым

12. NPC и враги в 3D

- добавление моделей врагов
- программирование движения неигровых персонажей
- знакомство с enum и состояниями
- изучение скриптов для преследования и атаки персонажа

Практическое задание: добавить новых врагов с различными характеристиками

13. Создание оружия и пуль

- взаимодействие с нодой RigidBody и создание физических объектов
- работа с нодой Area и ее событиями
- программирование подбора оружия
- программирование выстрела пуль

Практическое задание: добавить новый вид оружия

14. UI в 3D играх

- добавление пользовательского интерфейса для вывода количества пуль
- добавление собираемых предметов и вывод их количества на экран
- создание кнопки для выхода из игры
- создание начального меню

Практическое задание: улучшение внешнего вида UI

15. Добавление звукового оформления в игру

- скачивание и импорт звуковых файлов
- знакомство с элементами Audio
- активация звуков через скрипт
- синхронизация игры и звуков

Практическое задание: подключение новых звуков в игру

16. Визуальное оформление и работа со светом

- добавление системы частиц

- работа с нодами света
- знакомство со свойствами света
- наложение теней и работа со светом через скрипт

Практическое задание: улучшение внешнего вида игры

17. Подготовка к созданию хоррора

- добавление моделей
- работа с камерой
- взаимодействие элементов и кнопок в UI
- переключение между камерами

Практическое задание: скачивание и добавление в игру новых моделей

18. Создание хоррора

- добавление врагов
- анимация объектов
- знакомство с состояниями, анимациями и ключами
- скрипты для взаимодействия с анимациями

Практическое задание: подключение новых видов врагов в игру

19. Доработка хоррора

- создание начального меню
- добавление квестов
- знакомство с понятием signal
- программирование взаимодействий между скриптами через сигналы

Практическое задание: улучшение внешнего вида игры

20. Улучшение хоррора

- добавление ловушек
- создание сцены проигрыша и победы
- добавление звуковых эффектов и визуальных эффектов
- настройка света на уровне

Практическое задание: доработка проекта.

21. Подготовка к созданию игры Гонки

- добавление моделей машин
- создание Terrain
- добавление трасс и препятствий
- программирование движения машины

Практическое задание: скачивание и добавление новых моделей в игру

22. Физика и UI в гонках

- добавление таймера
- улучшение физики автомобиля
- добавление в игру погодных условий
- добавление моделей соперников

Практическое задание: добавление новых видов соперников

23. Поведение соперников и финиш

- программирование движения соперников
- знакомство с нодой PathFollow
- программирование финиша
- создание UI для вывода результатов

Практическое задание: улучшение внешнего вида игры

24. Улучшение игры гонки

- добавление меню с выбором машины
- программирование характеристик машин
- добавление начального меню
- улучшение пользовательского интерфейса

Практическое задание: улучшение дизайна игры.

25. Возвращение в 2D и создание игры Roguelike RPG

- подготовка уровней персонажа
- знакомство с анимацией в 2D
- изучение понятия «дерево анимаций»
- программирование движения персонажа

Практическое задание: добавление новых текстур в игру

26. Знакомство с Tileset'ами

- знакомство с понятием процедурной генерации
- добавление новых объектов
- изучение ноды YSort и расположения по слоям

Практическое задание: улучшение генератора и добавление туда новых элементов

27. Процедурная генерация

- добавление гранат и предметов, которые можно кидать
- звуковое сопровождение игры
- добавление частиц для улучшения визуальной составляющей

Практическое задание: улучшение внешнего вида игры

28. Добавление сундуков и собираемых предметов

- добавление сундуков
- создание пользовательского интерфейса
- добавление предметов
- написание скриптов для предметов

Практическое задание: добавление новых предметов

29. Добавление врагов и системы жизней

- создание врагов и программирование их движения
- создание программ для нанесения и получения урона
- изучение понятия Hitbox и Hurtbox
- добавление и программирование UI для вывода жизней

Практическое задание: улучшение характеристик врагов, добавление новых видов врагов

30. Новые локации

- создание новых сцен
- программирование перехода между уровнями
- дизайн новых уровней
- улучшение процедурной генерации

Практическое задание: добавление в игру новых уровней

31. Квесты

- знакомство с понятием квестов
- программирование квестов
- использование условий и сигналов для реализации квестов

Практическое задание: добавить новый квест

32. Экспорт игры и выкладывание на хостинг

- знакомство с хостингом
- регистрация и создание сайта
- экспорт игры в нужный формат
- загрузка игры на сайт, презентация

Практическое задание: выложить в открытый доступ все созданные за курс игры, презентация сайта как проекта

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

5.1. Контроль знаний, умений и навыков

Формой подведения итогов реализации дополнительной образовательной программы выступает текущая, промежуточная и итоговая аттестация.

Образовательный процесс осуществляется на основании учебного плана и регламентируется расписанием занятий для каждой учебной группы.

В целях оценки показателей знаний, умений и навыков обучающихся по дополнительной образовательной общеразвивающей программе "Разработка 2D и 3D игр на Python" проводится текущий и промежуточный контроль знаний, а также итоговая аттестация.

Виды текущего контроля:

- устный ответ на поставленный вопрос;
- проверка результатов выполнения практических заданий.

Виды промежуточного контроля:

- тестирование устное/письменное/с помощью электронных форм;
- проверка результатов выполнения практических работ/проектов по итогам учебного модуля.

Тестирование — это форма измерения знаний обучающихся, основанная на применении тестов. Материалы для промежуточного и итогового тестирования предоставляются вместе с комплектом учебно-методических материалов к программе.

Итоговая аттестация

Итоговая аттестация проводится с целью установления уровня знаний обучающихся с учетом прогнозируемых результатов обучения и требований к результатам освоения образовательной программы.

Итоговая аттестация обучающихся проводится в форме презентации проекта с демонстрацией результата – веб-сайта с играми, созданными в рамках данного курса.

Выдача обучающимся документов о дополнительном образовании (сертификат о прохождении курса) осуществляется при условии успешного прохождения итоговой аттестации.

5.2. Критерии оценивания освоения программы при проведении различных форм контроля:

Тестирование (Приложение 1. Примерные вопросы для промежуточного тестирования).

Процент результативности (правильных ответов при выполнении тестовых заданий):

Выполнение теста	Итоговая оценка
60% и более правильных ответов	"Зачтено"
Менее 60% правильных ответов	"Не зачтено"

Проверка выполнения практических работ (Приложение 2. Примерные задания для проверки усвоения качества учебного материала).

Система оценивания:

"Зачтено" – необходимый уровень выполнения задания достигнут, обучающийся демонстрирует хорошее знание теоретической и практической части материала занятия/учебного модуля, достигнуты промежуточные и/или итоговые результаты работы над заданием.

"Не зачтено" - необходимый результат/уровень освоения не достигнут, обучающийся не усвоил теоретические основы и/или изученные практические приемы программирования и инструменты Godot, не достиг промежуточных и итоговых результатов при выполнении задания.

Проверка результатов создания проекта на итоговой аттестации:

Критерии оценки	БАЛЛЫ
Обоснование выбора сюжета игры и его актуальность	0-2 балла
Реализовано не менее двух изученных типов игр	0-4 балла
Реализовано не менее двух уровней игры и портал между ними	0-4 балла
Реализованы скрипты для управления не менее двух персонажей (игрока и противника) в одной игре	0-4 балла
Реализовано игровое взаимодействие игрока с NPC	0-3 балла
Реализовано игровое взаимодействие игрока с предметами	0-3 балла
Показана самостоятельность работы над проектом	0-2 балла
Компетентность докладчика (ответы на вопросы)	0-2 балла
Итоговая оценка: «Не зачтено» «Зачтено»	0-14 баллов 15-23 баллов

6. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ (УЧЕБНИКИ, РАЗДАТОЧНЫЕ МАТЕРИАЛЫ, МЕТОДИЧЕСКИЕ УКАЗАНИЯ, ПЛАКАТЫ, СЛАЙДЫ, ИНТЕРНЕТ-РЕСУРСЫ)

6.1. Информационные и учебно-методические условия реализации программы

Учебно-методический комплект

Для реализации целей и задач обучения по общеобразовательной общеразвивающей программе "Разработка 2D и 3D игр на Python" используется комплект материалов преподавателя, который включает:

1. Текстовое методическое пособие с описанием целей, результатов каждого занятия, теоретического материала и практических работ.
2. Видеоурок для преподавателя с методическими указаниями и порядком объяснения учебного материала.
3. Раздаточный материал для учащихся - описание дополнительной самостоятельной работы учащихся по каждому занятию с примерами и рекомендациями по выполнению.
4. Описание мероприятий по контролю знаний – тестовые вопросы, практические задания.
5. Рекомендации по проведению итоговой аттестации и защиты проектов.
6. Дополнительные материалы – инструкции по установке необходимого программного обеспечения, описание технических требований к компьютерному оборудованию.

Материалы преподавателя размещаются на учебном портале преподавателей, размещенном на сервере информационно-телекоммуникационной сети "Интернет", и доступны по ссылке для всех преподавателей курса. Материалы обучающихся раздаются в печатном виде или рассылаются преподавателем индивидуально каждому обучающемуся.

6.2. Рекомендованная литература, рекомендованная для освоения программы:

Основная:

1. Широков А.И., Пышняк М. Информатика. Разработка программ на языке программирования Питон. Часть 1. Базовые языковые конструкции / А. И. Широков, Марина Пышняк – Москва: ИД МИСиС, 2020. – 144 с.
2. Мэтиз Э., Python. Карманный справочник/ Эрик Мэтиз – С.-Петербург., Питер, 2020 – 512 с.
3. Лутц М., Изучаем Python. Том 1 / Марк Лутц — Москва: Диалектика, 2019 – 832 с.
4. Лутц М., Изучаем Python. Том 2 / Марк Лутц — Москва: Вильямс, 2020 – 720 с.
5. Бриггс Джейсон. Python для детей. Самоучитель по программированию / Д.Бриггс. - Москва: Манн Иванов и Фербер (МИФ), 2016. – 289 с.

Дополнительная:

1. Лутц М., Python. Карманный справочник/ Марк Лутц — Москва: Диалектика, 2019 – 320 с.

2. Брайсон Пейн. Программирование на Python для детей и родителей / Б. Пейн. – Москва: Эксмо, 2017 – 354 с.
3. Buckley, D Godot Game Development for Beginners/ Daniel Buckley - Zenva Pty Ltd, 2020
4. Manzur A., Marques G., Godot Engine Game Development in 24 Hours, Sams Teach Yourself: The Official Guide to Godot 3.0/ Ariel Manzur, George Marques. – Pearson, 2018. - 665 с.

6.3. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", рекомендованных для освоения программы

- <https://godotengine.org/> - официальный сайт Godot
- <https://devdocs.io/godot~3.4/> руководство, документация по Python
- <https://habr.com/ru/post/531018/> Статья "Твоя первая игра на Godot Engine" @oleshka275
- <https://pythontutor.ru/> - проект Питонтьютор
- https://ru.hexlet.io/courses/python_101 – Хекслет. Курс «Введение в Python»
- <https://www.udemy.com/course/python-project/> - Python. Практика для начинающих
- https://docs.godotengine.org/en/3.0/getting_started/step_by_step/your_first_game.html
- <https://github.com> - веб-сервис для хостинга IT-проектов и их совместной разработки
- <https://habr.com/> - Портал habr.com

7. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

7.1. Материально-техническая и ресурсная база

Для реализации программы предполагается использование учебных аудиторий для проведения занятий лекционного типа, занятий семинарского типа, выполнения проектных работ, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации:

- Компьютерные классы, обеспечивающие доступ в Интернет, и оснащенные мультимедиа проектором или иными средствами визуализации учебного материала, магнитной доской или флипчартом.
- Электронный информационно-образовательный портал, размещенный на сервере в информационно-телекоммуникационной сети "Интернет".
- Специальное программное обеспечение для разработки, необходимое для реализации образовательных задач курса.
- Стандартное программное обеспечение для работы над разработкой учебно-методических материалов.
- Мастерские и аудитории для проведения открытых занятий.

Специальных помещений, предполагающих наличие какого-либо специального оборудования для реализации данной программы, не предусматривается.

7.2. Кадровое обеспечение программы

Образовательный процесс по программе осуществляется педагогом дополнительного образования с профильным высшим или средним профессиональным образованием.

К занятию педагогической деятельностью по дополнительной общеобразовательной программе также допускаются лица, обучающиеся по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующим направленности дополнительных общеобразовательных программ, и успешно прошедшие промежуточную аттестацию не менее чем за два года обучения.

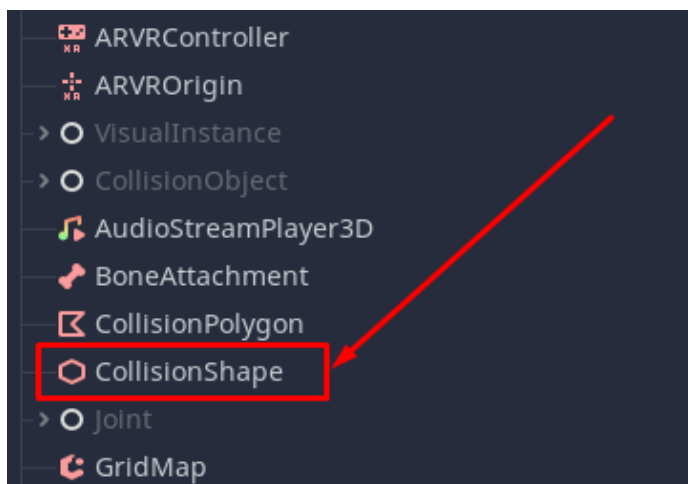
Реализация дополнительной общеобразовательной (общеразвивающей) программы обеспечивается руководящими и педагогическими работниками организации, а также лицами, привлекаемыми к реализации программы на условиях гражданско-правового договора.

У педагогического работника, реализующего дополнительную общеобразовательную программу, должны быть сформированы основные компетенции, необходимые для обеспечения успешного достижения обучающимися планируемых результатов освоения программы, в том числе умения:

- обеспечивать условия для успешной деятельности, позитивной мотивации, а также самомотивирования обучающихся;
- осуществлять самостоятельный поиск и анализ информации с помощью современных информационно-поисковых технологий;
- разрабатывать программы учебных предметов, выбирать учебники и учебно-методическую литературу, рекомендовать обучающимся дополнительные источники информации, в том числе Интернет-ресурсы;
- реализовывать педагогическое оценивание деятельности обучающихся;
- работать с текстовыми редакторами, электронными таблицами, электронной почтой и браузерами, мультимедийным оборудованием.

Приложение 1. Примерные вопросы для промежуточного тестирования

1. Для чего нужен компонент, изображенный на скриншоте?



- Для создания 3D-элементов
- **Для добавления твердости объекту**
- Для движения персонажа
- Для создания физики персонажа

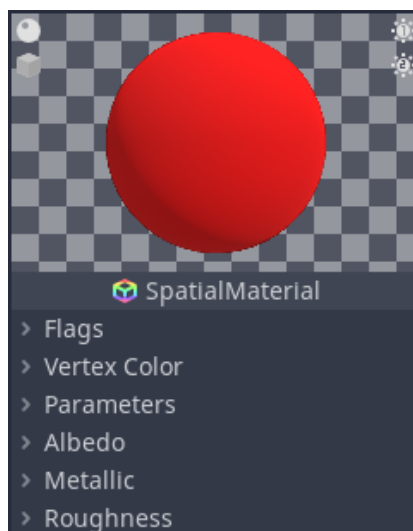
2. Что делает функция, изображенная на скриншоте?

```
func change_scene(path):  
>| get_tree().change_scene(path)
```

- Меняет цвет фона
- **Меняет уровень игры**
- Меняет внешний вид объекта
- Меняет настройки игры

3. В каком параметре из указанных на скриншоте можно поменять цвет объекта?

- Flags
- Vertex Color
- Parameters
- **Albedo**
- Metallic
- Roughness



4. Какая команда нужна для удаления объекта с уровня?

- potato_free()
- **queue_free()**
- memory_free()
- delete()

5. Какой сигнал проверяет вхождение объекта в Area?

- **body_entered**
- body_exited
- mouse_entered
- ready

6. Какой элемент нужен для вывода надписи на экран?

- Text
- TextArea
- EditText
- **Label**

7. Когда срабатывает эта функция?

```
func _process(delta):  
>|   move_state(delta)
```

- При запуске игры
- При столкновении двух объектов
- **Каждый кадр**
- При выходе из игры

8. Что значит слово “export” в данной строке?

```
export(String) var action
```

- **Переменная доступна для редактирования в редакторе свойств**
- Переменная объявлена публичной
- Переменную можно использовать в других скриптах
- Для создания константы

9. Для чего нужен Timer?

- **Чтобы сделать какие-то действия через определённое время**
- Чтобы показывать время, проведённое в игре
- Для завершения игры
- Для показа изображения

10. С помощью каких элементов сделаны кнопки на скриншоте?



- Control
- TextEdit
- **TextureButton**
- Container

11. За что отвечает свойство Scale?

- Положение
- Поворот
- **Размер**

12. Какая комбинация клавиш создает дубликат узла?

- ctrl+w
- **ctrl+d**
- shift+d
- ctrl+t

13. Что такое KinematicBody2D?

- Узел для персонажа (чтобы он двигался)
- Узел для твёрдых предметов
- Узел для создания картинке
- Узел для создания твердости

14. Для создания твердых объектов мы использовали узел...

- Body
- RigidBody
- KinematicBody
- **StaticBody**

15. Какая команда проверяет, что мы находимся на земле?

- **is_on_floor**
- Stay
- is_on_wall
- on_floor

16. Какая из этих функций срабатывает каждый кадр?

- **_process**
- _get_tree
- _exit
- _ready

17. С помощью какой команды мы получаем значение с клавиатуры?

- **Input.get_action_strength**
- Input.get_doing_strength
- Input.set_doing_strength
- Input.get_connect_strength

18. Какая команда перемещает персонажа по заданному вектору?

- **move_and_slide**
- moving
- move_and_drag
- move

19. Как проверить, что в портал вошёл именно персонаж?

- **body.name == "Player"**
- body == "Player"
- body.name = "Player"
- body.name == "KinematicBody"

20. С помощью какой команды мы двигаем персонажа влево?

- **moving.x -= speed * delta**
- moving.x += speed * delta
- moving.x = speed * delta
- moving.z -= speed * delta

21. Какая команда используется для смены уровня?

- `get_tree.change_scene`
- `change_level`
- `get_tree.change_level`
- `change_scene`

22. Какой узел позволяет реализовать ландшафт?

- **HTerrain**
- Landscape
- Scenery
- View

23. Как сделать из модели динамический объект (т.е. объект, который можно было бы двигать)?

- Сделать объект `DynamicObject` и положить в него модель
- Включить у модели параметр `Rigid`
- **Сделать объект `Rigidbody` и положить в него модель**
- Включить у модели параметр `Dynamic`

24. Какое действие выполняет команда `bullet.instance()`?

- Создает копию объекта `bullet`
- Уничтожает объект `bullet`
- Перемещает объект `bullet`
- Переворачивает объект `bullet`

25. Какой узел является основным для настройки анимаций у объекта?

- `Animator`
- **`AnimationPlayer`**
- `AnimationController`
- `Anim`

26. Что делает команда `look_at`?

- Позволяет приближать камеру
- Позволяет смотреть внутрь объекта
- **Поворачивает один объект "лицевой стороной" к другому**
- Включает камеру в другом объекте

27. Какая команда позволяет перезагружать сцену в Godot?

- `get_tree().reload_current_scene()`
- `get_tree().reboot_scene()`
- `get_tree().restart_scene()`
- `get_tree().rebirth_current_scene()`

28. Когда срабатывает функция `finished` у узла `AudioStreamPlayer`?

- когда музыка заканчивает проигрывание
- когда музыка начала проигрывание
- когда игрок нажимает кнопку остановки музыки.

Задание 1. Закрепление навыков работы со скриптами

- Создать еще одного неигрового персонажа (NPC) с другим заданием, который при выполнении даёт другой предмет, отличный от использованного на занятии
- Добавить возможность перемещаться новым NPC
- Создать префабы NPC для переноса на другие уровни

Образец реализации:

```
1 extends KinematicBody2D
2
3 onready var UI = get_parent().get_node("CanvasLayer/Panel")
4
5 export var dialog = {"ready": "Найди для меня 3 монеты!", "end": "Молодец! Держи яблоко!"}
6
7 func _on_Area2D_body_entered(body):
8     if body.name == "Player":
9         UI.visible = true
10        if GameManager.coins < 3:
11            UI.get_node("Label").text = dialog["ready"]
12        else:
13            UI.get_node("Label").text = dialog["end"]
14            var item = load("res://Prefabs/Item.tscn")
15            var item_instance = item.instance()
16            get_parent().add_child(item_instance)
17            item_instance.position = position + Vector2(300, 0)
18            GameManager.coins -= 3
19
20 func _on_Area2D_body_exited(body):
21     if body.name == "Player":
22         UI.visible = false
```

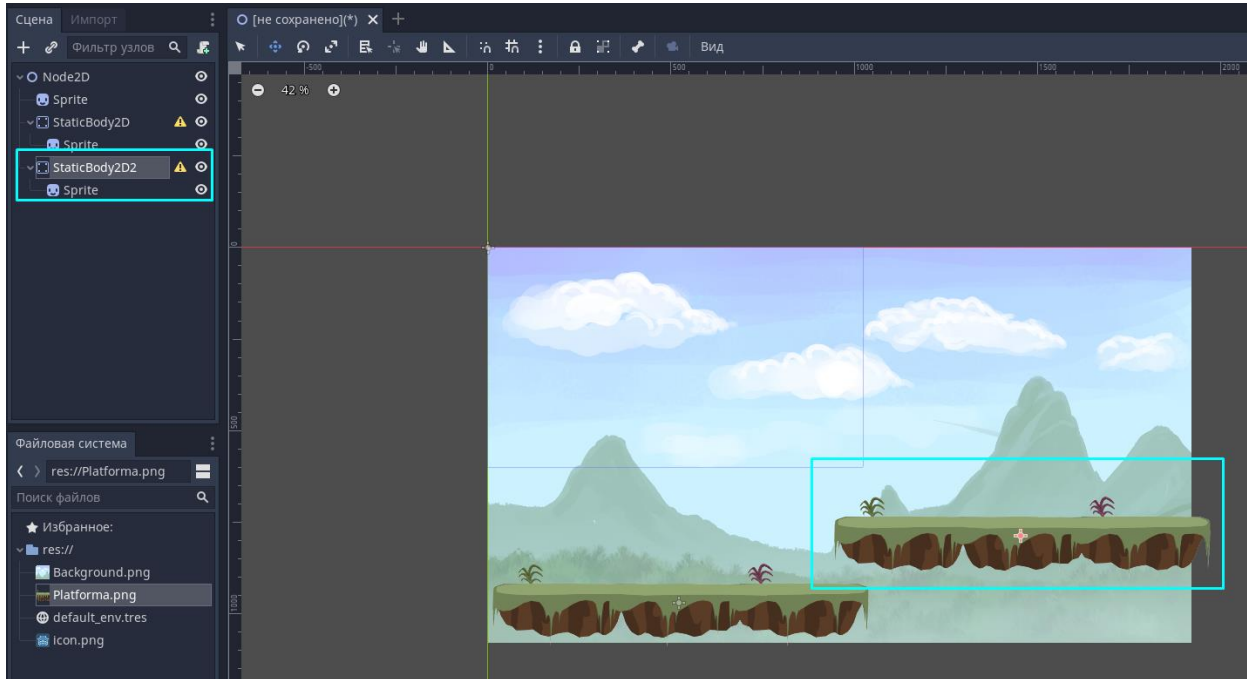
```
1 extends KinematicBody2D
2
3 export var min_x = 0
4 export var max_x = 0
5 export var speed = 50
6
7 var moving = Vector2.ZERO
8 var direction = "left"
9
10 func _physics_process(delta):
11     if position.x <= min_x:
12         direction = "right"
13     elif position.x >= max_x:
14         direction = "left"
15
16     if direction == "left":
17         moving.x -= speed * delta
18     elif direction == "right":
19         moving.x += speed * delta
20
21     moving.x = clamp(moving.x, -speed, speed)
22
23     moving = move_and_slide(moving, Vector2.UP)
```

Задание 2. Повторение и закрепление навыков работы с интерфейсом Godot

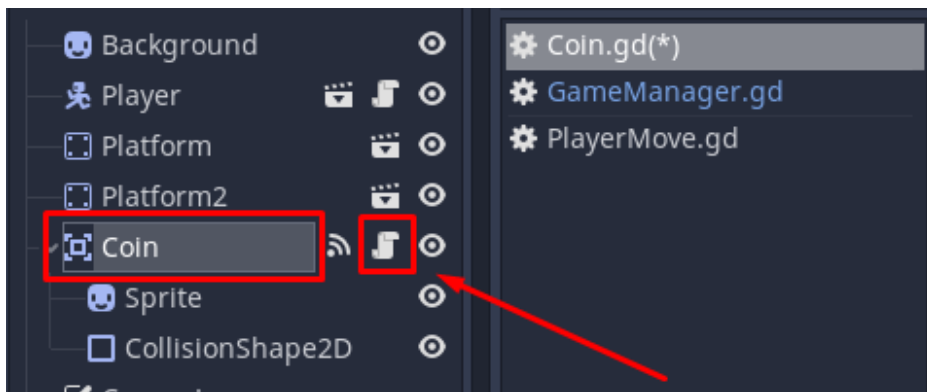
Создать второй уровень в игре, сделать новый фон, расставить платформы и монеты.

Образец плана реализации:

1. Фон. Node2D+Sprite
2. Платформа. StaticBody2D(=Platform)+Sprite – 3 шт.



3. Монета. Area2D(=Coin)+Sprite+CollisionShape2D+Сигнал+Script



```
2
3 > func _on_Coin_body_entered(body):
4 >   GameManager.coins += 1
5 >   queue_free()
6 |
```

Задание 3. Создание зелий

По аналогии со скриптами, созданными на занятии, самостоятельно создайте следующие зелья:

- Создать зелье, восстанавливающее жизни
- Создать зелье, наносящее урон персонажу

Задание 4. Увеличение монет в игре

Вам необходимо увеличить количество монет у игрока. Напишите скрипт для этого действия, проверьте разные способы написания увеличения количества монет. Ответьте, какую из следующих записей можно использовать для этого действия:

- `GameManager.coins += 1`
- `GameManager += 1`
- `GameManager = 1`
- `GameManager.coins = 1`

Задание 5. Работа со скриптами

Запишите скрипт, который работает с изменением скорости движения персонажа. Добавьте в код комментарии, поясняющие все строки кода:

```
1 extends Area2D
2
3 func _on_PotionSpeed_body_exited(body):
4     >| if body.name == "Player":
5         >| >| body.speed += 50
6         >| >| queue_free( )
```

Задание 6. Настройка UI проекта

Вы уже знаете, что Godot ограничен на работу со встроенными шрифтами. Расширьте набор шрифтов для своего проекта. Выберите и скачайте новый шрифт с <https://fonts.google.com> и настройте его использование в проекте.